Initially, the applicants note that linking claims 15-18 were added in the Response to Restriction Requirement filed August 1, 2002. The Office Action states that since these added linking claims were not presented when the restriction requirement was made, they are therefore grouped with the non-elected claims of Group II (claims 7-13). The applicants respectfully disagree with this analysis and grouping associated with claims 15-18.

Claims 15-18 recite features similar to those in original claims 1-4 (in means plus function format). As such, the applicants believe that claims 15-18 would be properly classified with the Group I claims (claims 1-6 and 14), which were elected in the Response to Restriction Requirement. Therefore, the applicants respectfully request that these claims be examined in the next communication. In the event that claims 15-18 are not examined, the applicants respectfully request that the Examiner provide an explanation as to why claims 15-18 would not be included with the claims of Group I.

Claims 4 and 18 have been objected to for minor informalities. The applicants note that claim 18, as discussed above, was not substantively examined in the current Office Action. In any event, the Office Action states that the phrase "first and second current instruction" in claims 4 and 18 should be changed to "first and second current instructions." Claims 4 and 18 have been amended in accordance with the Examiner's suggestion.

Claim 14 has been rejected under 35 U.S.C. § 112, second paragraph as being indefinite. In particular, the phrases "the computer graphic system" and "the single copy" were considered to lack proper antecedent basis. Claim 14 has hereby been

B

amended and is now believed to be definite. Accordingly, withdrawal of the rejection of claim 14 is respectfully requested.

Claims 1-3 and 14 have been rejected under 35 U.S.C. § 103 (a) as being unpatentable over Burgess (U.S. Patent 5,652,888) in view of "Design Patterns, Elements of Reusable Object-Oriented Software," pages 127-134 by Erich Gamma et al. (hereinafter Gamma). The rejection is respectfully traversed.

Claim 1 recites generating a first command control vector for a first input message, the first command control vector identifying a method object that contains one or more instructions for processing the first input message. Claim 1, as amended, also recites that the generating a first command control vector comprises identifying the method object in the first command control vector and identifying, in the first command control vector, a first current instruction of the method object, wherein the first current instruction is used to process the first input message. This feature was previously recited in claim 2.

The Office Action states that Burgess discloses these features and points to col. 4, lines 19-54 and col. 7, lines 3-54 for support (Office Action – page 4). Apparently, the Office Action considers the pointer to the target member function in Burgess to be equivalent to identifying a first current instruction of the method object, as recited in claim 1. The applicants respectfully disagree.

Burgess at col. 4, lines 19-54 discloses that message information is encapsulated in an object of an event class. Each class of an event object contains information that is passed to a target member function according to the function prototype (Burgess – col. 4, lines 33-35). This portion of Burgess does not disclose identifying a first current

instruction of method object that is identified in a command control vector, as recited in claim 1.

Burgess at col. 7, lines 3-54 refers to Fig. 7, which illustrates the flow of control of sending a message from a source object to a target object. In step 721, the member function determines which entries in the connection array correspond to the source member function. When an entry is found, "the member function InformTargets invokes the member function NotifyEvent of the target object passing a pointer to the event, a pointer to the source object, and a pointer to the target member function" (col. 7, lines 12-18). Passing pointers in this manner to the target member function is not equivalent to identifying a first current instruction of method object that is identified in a command control vector, as recited in claim 1.

Claim 1 also recites identifying the same method object in the second command control vector and identifying, in the second command control vector, a second current instruction of the method object, wherein the second current instruction is used to process the second input message. As to this feature, the Office Action merely states that Burgess "uses this logic to process all messages/command control vectors, including the first and second" (Office Action – page 4). Even if, for the sake of argument, Burgess used the same logic to process all messages, this would not read on the feature in claim 1. That is, Burgess does not disclose identifying any instruction of a method object, much less an instruction used to process the second input message, as recited in claim 1. Gamma also does not disclose or suggest these features.

For at least these reasons, the combination of Burgess and Gamma does not disclose each of the features of claim 1.

In addition, even if, for the sake of argument, the combination of Burgess and Gamma was construed to disclose each of the features of claim 1, the Office Action does not provide the motivation required under 35 U.S.C. § 103 as to why it would have been obvious to combine these references.

The Office Action states that Gamma discloses creating a class to provide only a single copy of the class object (class having one instance). The Office Action then states that given the teaching of Gamma, it would have been obvious to provide a single copy of the method object (window manager) of Burgess (Office Action – page 4). This statement points to no portion of either reference as providing objective motivation for combining Burgess with Gamma. The mere fact that one reference allegedly provides some missing disclosure with respect to a claim does not satisfy the requirements of 35 U.S.C. § 103 as to why it would have been obvious to combine these references.

For at least the reasons discussed above, withdrawal of the rejection and allowance of claim 1 are respectfully requested.

Claims 2 and 3 depend from claim 1 and are believed to be allowable for at least the reasons claim 1 is allowable. In addition, these claims recite additional features neither disclosed nor suggested by either Burgess or Gamma.

For example, claim 2 recites identifying, in the first command control vector, a communication link from which the first input message is received and identifying, in the first command control vector, a destination device for which the first input message is intended. The Office Action states that Burgess discloses a pointer to the source object and a pointer to the target object (Office Action – page 4). These pointers in Burgess are apparently considered to be equivalent to the claimed identifying of the communication

link from which the first input message is received and identifying a destination device for which the first input message is intended. The applicants respectfully disagree.

Burgess may use pointers to the source object and target object. The pointers in Burgess, however, do not identify a communication link from which a message is received or a destination device to which the message is intended, as recited in claim 2.

Claim 3 recites similar features as claim 2 with respect to the second input message. Similar to the discussion with respect to claim 2, the pointers in Burgess do not identify a communication link from which a message is received or a destination device to which the message is intended, as recited in claim 3.

For at least these additional reasons, withdrawal of the rejection and allowance of claims 2 and 3 are respectfully requested.

Claim 4 has been rejected under 35 U.S.C. § 103 (a) as being unpatentable over Burgess in view of Gamma and further in view of admitted prior art. The rejection is respectfully traversed.

Claim 4 is dependent on claim 3 and is believed to be allowable for at least the reasons claim 3 is allowable. In addition, claim 4 recites additional features neither disclosed nor suggested by either Burgess, Gamma or the admitted prior art.

For example, claim 4 recites that the processing the first and second input messages comprises using a single copy of a script to process the first and second input messages. The Office Action states that the admitted prior art discloses a method object invoking a script and points to page 2, line 29 of the applicants' specification for support (Office Action – page 5).

The applicants' specification does disclose that method objects may be, for example, a script. This disclosure, however, does not read on the feature recited in claim 4. The Office Action further states that the combined teachings of Burgess, Gamma and the admitted prior art would then provide a single copy of the script. The applicants respectfully disagree.

None of the references, taken alone, or in combination discloses the use of first and second instructions that both invoke a single copy of the script to process first and second input messages. Therefore, even if the admitted prior art was combined with Burgess and Gamma, the claimed invention would not result.

For at least these additional reasons, withdrawal of the rejection and allowance of claim 4 are respectfully requested.

Claims 5 and 6 have been rejected under 35 U.S.C. § 103 (a) as being unpatentable over Burgess in view of Gamma and the admitted prior art and further in view of "Compiling Distributed C++," by H. Carr et al. (hereinafter Carr). The rejection is respectfully traversed.

Claims 5 and 6 are dependent on claims 1 and 4, respectively, and are believed to be allowable for at least the reasons claims 1 and 4 are allowable. In addition, claim recites additional features neither disclosed nor suggested by either Burgess, Gamma, the admitted prior art or Carr.

For example, the Office Action admits that Burgess does not disclose storing, in a first data object, data that is generated during execution of the script for the first command control vector; and storing, in a second data object, data that is generated during execution of the script for the second command control vector, as recited in claim

5. The Office Action, however, states that Carr discloses using a data object (value object) to store data generated during execution (return values) of a C++ program and points to pages 499-500 of Carr for support (Office Action – page 5). Even if Carr discloses such features, such a disclosure is not equivalent to storing, in a first data object, data that is generated during execution of the script for the first command control vector or storing, in a second data object, data that is generated during execution of the script for the second command control vector, as recited in claim 5.

Therefore, even if Carr was combined with the combination of Burgess, Gamma and the admitted prior art, the claimed invention would not result.

In addition, even if, for the sake of argument, the combination of Burgess, Gamma, the admitted prior art and Carr was construed to disclose each of the features of claim 5, the motivation to combine Carr with the other disclosures does not satisfy the requirements of 35 U.S.C. § 103. For example, the Office Action states that it would have been obvious to combine the teachings of Burgess as modified by Gamma and Carr because the former implements the teaching in language C++ and the latter details one version of the language C++ (Office Action – pages 5-6). This statement is merely a conclusory statement and no portion of any of the references is pointed to as providing objective motivation for combining these disclosures. The mere fact that these references involve the C++ programming language does not provide the required motivation under § 103 as to why one of ordinary skill in the art would combine these references.

For at least these reasons, withdrawal of the rejection and allowance of claim 5 are respectfully requested.

As to claim 6, the applicant notes that this claim was rejected based on the combination of Burgess, Gamma, the admitted prior art and Carr. Carr, however, was not relied upon in the grounds of rejection. Therefore, clarification as to the grounds of rejection is respectfully requested.

In any event, the Office Action states that the admitted prior art disclose processing n logic units of instructions of a first type, interrupting such processing and processing m logical units of instructions for a second type and points to page 3, lines 3-8 of the applicant's specification for support. The applicants respectfully disagree.

The applicants' specification at page 3, lines 3-8 refers to multi-tasking systems which employ preemptive time-slice processing. Tasks in such systems are automatically interrupted after a predetermined period of time or after a specific instruction type. Processing may resume at a later point, preferably form where it was interrupted. This disclosure in the applicants' specification at page 3 is not equivalent to processing a number *n* of logical units of instructions for the first command control vector; interrupting processing of the first command control vector; and processing a number *m* of logical units of instructions for the second command control vector.

Therefore, even if the admitted prior art was combined with Burgess and Gamma, the claimed invention would not result. For at least this additional reason, withdrawal of the rejection and allowance of claim 6 are respectfully requested.

CONCLUSION

In view of the foregoing amendments and remarks, the applicants respectfully request withdrawal of the outstanding rejections and the timely allowance of this application.

To the extent necessary, a petition for an extension of time under 37 C.F.R. § 1.136 is hereby made. Please charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, to Deposit Account 13-2491 and please credit any excess fees to such deposit account.

Respectfully submitted,

HARRITY & SNYDER, L.L.P.

By: _____
Glenn Snyder
Reg. No. 41,428

Date: January 23, 2003

11240 Waples Mill Road
Suite 300
Fairfax, VA 22030
Telephone: (571) 432-0800
Facsimile: (571) 432-0808

MARKED-UP VERSION OF AMENDMENT SHOWING CHANGES MADE IN THE SPECIFICATION:

The paragraphs at page 1, line 13 to page 2, line 22 have been amended as follows:

1. U.S. Patent Application titled, "System and Method for Emulating Telecommunication Network Devices," Serial No. [(to be assigned), Attorney Docket No. COS-94-027 (1575.1770000),] 08/987,229, filed December 9, 1997, by John V. McLain, Jr., and Damon Curnell, [filed concurrently herewith] now U.S. Patent No. 6,295,518;

2. U.S. Patent Application titled, "System and Method for Performing Hybrid Preemptive and Cooperative Multi-Tasking in a Computer System," Serial No. [(to be assigned), Attorney Docket No. COS-97-032 (1575.2610000),] 08/987,633, filed December 9, 1997, by John V. McLain, Jr., and Damon Curnell, [filed concurrently herewith] now U.S. Patent No. 6,256,659;

3. U.S. Patent Application titled, "System and Method for Generating Responses to Inputs Using a Hybrid State Engine Table," Serial No. [(to be assigned), Attorney Docket No. COS-97-034 (1575.2630000),] 08/987,850, filed December 9, 1997, by John V. McLain, Jr., and Damon Curnell, [filed concurrently herewith] now U.S. Patent No. 5,974,532;

B

4. U.S. Patent Application titled, "Method and Apparatus for Emulating a Dynamically Configured Digital Cross-Connect Network," Serial No. [(to be assigned), Attorney Docket No. COS-94-029 (1575.0690000),] 08/641,458, by John V. McLain, Jr., and James Dellinger, filed May 1, 1996, now U.S. Patent No. 5,809,286;

5. U.S. Patent Application titled, "Method and Apparatus for Emulating a Digital Cross-Connect Network," Serial No. [(to be assigned), Attorney Docket No. COS-94-021 (1575.0700000),] 08/641,459, by John V. McLain, Jr., filed May 1, 1996, now U.S. Patent No. 5,748,617;

6. U.S. Patent Application titled, "Method and Apparatus for Emulating Digital Cross-Connect Network using a Flexible Topology to Test MCS Network Management," Serial No. [(to be assigned), Attorney Docket No. COS-95-013 (1575.0880000),] 08/641,461, by John V. McLain, Jr., filed May 1, 1996, now U.S. Patent No. 5,867,689;

7. U.S. Patent Application titled, "Method and Apparatus for Emulating a Network of State Monitoring Devices," Serial No. [(08/672,141), Attorney Docket No. COS-94-020, (1575.1010000),] 08/672,141, by John V. McLain, Jr., filed June 27, 1996, now U.S. Patent No. 5,812,826;

B

8. U.S. Patent Application titled, "Method and Apparatus for Simulating Multi-Tasking," Serial Number [(to be assigned), Attorney Docket No. COS-94-030,] No. 08/641,460, filed May 1, 1996, by John V. McLain, Jr., now U.S. Patent No. 5,850,536;

9. U.S. Patent Application titled, "System, Method and Computer Program product for Digital Cross Connect Testing," Serial Number [(to be assigned), Attorney Docket No. COS-96-006, (1575.1710000),] No. 08/774,650, by John V. McLain, Jr., and Dale W. Harris, filed December 30, 1996, now U.S. Patent No. 5,954,829; and

10. U.S. Patent Application titled, "Digital Cross Connect Command Script Generator," Serial Number 08/774,651, [Attorney Docket No. COS-96-042,] by John V. McLain, Jr. filed December 31, 1996, now U.S. Patent No. 5,854,930.

IN THE CLAIMS:

Claims 1-6, 14 and 18 have been amended as follows.

1. (Amended) A method for managing computer system resources, comprising [the steps of]:

[(1)] generating a first command control vector for a first input message, the first command control vector identifying a method object that contains one or more instructions for processing the first input message, wherein the generating a first command control vector comprises:

identifying the method object in the first command control vector, and

identifying, in the first command control vector, a first current instruction of the method object, wherein the first current instruction is used to process the first input message;

[(2)] generating a second command control vector associated with a second input message, the second command control vector identifying the same method object identified by the first command control vector, the method object containing one or more instructions for processing the second input message, wherein the generating a second command control vector comprises:

identifying the same method object in the second command control vector; and

identifying, in the second command control vector, a second current instruction of the method object, wherein the second current instruction is used to process the second input message;

[(3)] providing a single copy of the method object for the first and second command control vectors; and

[(4)] processing the first and second input messages using the single copy of the method object.

2. (Amended) The method according to claim 1, wherein [step (1)] the generating a first command control vector further comprises:

[(a)] identifying, in the first command control vector, a communication link from which the first input message is received; and

[(b)] identifying, in the first command control vector, a destination device for which the first input message is intended[;

(c) identifying the method object in first command control vector;

(d) identifying, in the first command control vector, a first current instruction of the method object, wherein the first current instruction is used to process the first input message].

3. (Amended) The method according to claim 2, wherein [step (2)] the generating a second command control vector further comprises:

[(a)] identifying, in the second command control vector, a communication link from which the second input message is received; and

[(b)] identifying, in the second command control vector, a destination device for which the second input message is intended[;

(c) identifying the same method object in second command control vector; and

(d) identifying, in the second command control vector, a second current instruction of the method object, wherein the second current instruction is used to process the second input message].

4. (Amended) The method of claim 3, wherein the first and second current [instruction] instructions are the same instruction and the same instruction invokes a script, wherein [step (4) comprises the step of] the processing the first and second input messages comprises:

[(a)] using a single copy of a script to process the first and second input messages.

5. (Amended) The method of claim 4, wherein [step (4)(a)] the using a single copy of the script comprises:

[(i)] identifying current script instructions in the first and second command control vectors for processing the first and second input messages, respectively;

[(ii)] storing, in a first data object, data that is generated during execution of the script for the first command control vector; and

[(iii)] storing, in a second data object, data that is generated during execution of the script for the second command control vector.

6. (Amended) The method according to claim 1, wherein [step (4) comprises the steps of] the processing the first and second input messages comprises:

[(a)] processing a number *n* of logical units of instructions for the first command control vector;

[(b)] interrupting processing of the first command control vector; and

[(c)] processing a number *m* of logical units of instructions for the second command control vector.

14. (Amended) A computer program product for permitting a computer system to manage computer system [resource] resources, said computer program product comprising:

a computer usable medium having computer readable program code means embodied in said medium for causing an application program to execute on the computer [graphics] system, said computer readable program code means comprising:

a computer readable first program code means for causing the computer system to generate a first command control vector for a first input message, the first command control vector identifying a method object that contains one or more instructions for processing the first input message, the first program code means further causing the computer system to identify a first current instruction of the method object and use the first current instruction to process the first input message;

a computer readable second program code means for causing the computer system to generate a second command control vector associated with a second input message, the second command control vector identifying the same method object identified by the first command control vector, the method object containing one or more instructions for processing the second input message, the second program code means further causing the computer system to identify a second current instruction of the method object and use the second current instruction to process the second input message; and

a computer readable third program code means for causing the computer system to process the first and second input messages using [the] a single copy of the method object.

18. (Amended) The system of claim 17, wherein the first and second current [instruction] instructions are the same instruction and the same instruction invokes a

script, wherein the means for processing the first and second input messages using the single copy of the method object further comprises:

means for using a single copy of a script to process the first and second input messages.